

Традиционный подход к написанию параллельных программ состоит в применении последовательно-параллельной схемы управления вычислениями. При этом каждый шаг вычислений состоит из фазы локальных вычислений, фазы коммуникаций и барьерной синхронизации. Такой подход прост в реализации и позволяет получить параметрические оценки эффективности практически для всех известных параллельных архитектур [1]. Недостатком является низкая эффективность при близких значениях времен вычислений и коммуникаций. Такая ситуация характерна для распределенных вычислений с использованием типовых рабочих станций в локальных и глобальных сетях (GRID-вычислений). Для GRID-приложений путем совмещения фаз коммуникаций и вычислений, а также путем асинхронного управления по готовности данных без барьерной синхронизации, удается достичь высокой эффективности. Сложность построения теоретических оценок эффективности таких алгоритмов, особенно с учетом случайного поведения коммуникационной среды, делает актуальным разработку методов дискретно-событийного моделирования вычислений на стадии проектирования распределенных вычислительных алгоритмов.

В качестве базы имитационной модели был использован визуальный язык GraphPlus [2]. Он позволяет строить обобщенные описания вычислительных процессов, абстрагируясь от конкретного алгоритма. Этот прием известен как построение каркасов приложений [3]. Исследовался каркас для решения явных и явно-неявных разностных схем при одномерной декомпозиции области данных. Схема взаимодействия процессов в таком каркасе представляет собой цепочку из попарно взаимодействующих процессов. Модель имеет 4 параметра: время передачи сообщения между процессами, время вычисления правой и левой границы сегмента данных и время вычисления собственно сегмента данных. Во всех экспериментах время вычисления границ составляло 0,5 единиц; время вычисления сегмента – 5 единиц; процесс останавливался после 1000 итераций.

В экспериментах исследовалось влияние времени задержки передачи сообщения на эффективность. На основе анализа пространственно-временных диаграмм процессов, а также графика изменения эффективности вычислений во времени было установлено предельное время передачи сообщений, при котором достигается 100% эффективность в стационарном режиме. Оно составило  $\frac{1}{2}$  от времени вычисления сегмента. Это подтверждает корректность разработанного распределенного алгоритма. Была проанализирована зависимость времени вычислений и эффективности от характера распределения и длительности задержки передачи сообщений между процессами (джиттера). На рис.1 и рис.2 показаны результаты модельных экспериментов. Ромбами на графиках отмечены кривые для постоянных времен задержек, кружками – кривые задержек, распределенных по равномерному закону. Из графиков можно заключить, что до определенного значения (равного  $\frac{1}{2}$  от времени вычисления сегмента) как величина, так и характер распределения времен задержек не оказывают существенного влияния на эффективность.

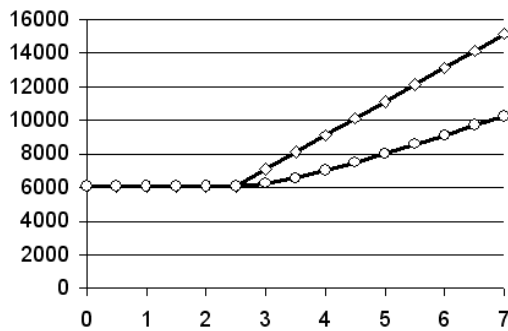


Рис.1. Зависимость времен выполнения 1000 циклов на конвейере из 12 процессов от времени передачи сообщений между процессами.

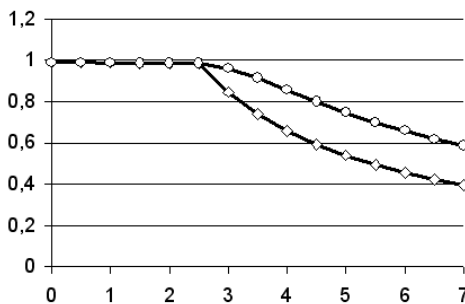


Рис.2. Зависимость эффективности вычислений при выполнении 1000 циклов на конвейере из 12 процессов от времени передачи сообщений между процессами.

Таким образом, нами экспериментально доказано:

1) благодаря совмещению во времени вычислений и коммуникаций, исследуемая модель применима для GRID-вычислений;

2) у вычислительной модели имеется «интервал устойчивости», в котором качество обслуживания (QoS) коммуникационной среды незначительно влияет на эффективность.

#### ЛИТЕРАТУРА

1. Skillicorn D.B., Hill J., McColl B. Questions and answers about BSP // Scientific programming. 1997. 6, N3, pp.249-274.
2. Востокин С.В. Технология моделирования распределенных систем, основанная на визуальном языке и ее приложения – Известия СНЦ РАН, том 6, №1(10) 2004, с.185-193.
3. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – СПб: Питер, 2003.–368с.